A Privacy-Preserving Social P2P Infrastructure for People-Centric Sensing

Michael Dürr and Kevin Wiesner

Mobile and Distributed Systems Group, Institute for Informatics
Ludwig-Maximilians-Universität München, Munich, Germany
{michael.duerr,kevin.wiesner}@ifi.lmu.de

Abstract

The rapid miniaturization and integration of sensor technologies into mobile Internet devices combined with *Online Social Networks* allows for enhanced sensor information querying, subscription, and task placement within *People-Centric Sensing* networks. However, PCS systems which exploit knowledge about OSN user profiles and context information for enhanced service provision might cause an unsolicited application and dissemination of highly personal and sensitive data.

In this paper, we propose a protocol extension to our OSN design *Vegas* which enables secure, privacy-preserving, and trustful P2P communication between PCS participants. By securing knowledge about social links with standard public key cryptography, we achieve a degree of anonymity at a trust level which is almost good as that provided by a centralized trusted third party.

Keywords and phrases People-Centric Sensing, Online Social Networks, P2P, Privacy, Trust

Digital Object Identifier 10.4230/OASIcs.KiVS.2011.176

1 Introduction

The increasing spread of powerful mobile Internet devices like smartphones or tablets, accompanied by their rich set of integrated sensing facilities, already allow for numerous mobile and context-enriched applications. Induced by those technical innovations, People-Centric Sensing (PCS), a recently emerging research area in the field of Wireless Sensor Networks, attracts increasing attention. PCS focuses on the collection of user-generated sensor data as well as its application-oriented aggregation and utilization at Internet-scale. In contrast to traditional sensor networks, PCS heavily builds on sensor information which is generated by a user's personal sensing environment [3, 5, 7]. Triggered by the rocketing number of *Online Social Network* (OSN) users and the imminent integration of several mobile sensor devices into personal sensing environments, recent research attempts to combine PCS environments and OSNs [1, 8]. Knowledge derived from social graphs and OSN user profile information can significantly improve the process of sensor information querying, sensor data subscription, and sensor task placement.

A well-known problem of PCS environments and sensor data context information generation emerges from the insufficient compliance with user privacy and security demands. Prevailing concerns stem from the possibility that any other subscriber could infer user profiles from continuous tracking information in order to e.g. perform unsolicited advertisement or even personal attacks. Despite recent efforts to incorporate privacy and security features into PCS systems [3, 5, 7] and OSNs [2, 4], present architectures still do not provide adequate precautions to fulfill all related privacy and security demands. On the one hand, experiences from the operation of location-based services have shown that a widespread utilization of such systems suffers from inadequate anonymization. The same problem arises in association with PCS networks as users contribute sensing data comprising personal information which

17th GI/ITG Conference on Communication in Distributed Systems (KiVS'11).
Editors: Norbert Luttenberger, Hagen Peters; pp. 176–181

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

must not be publicly available. On the other hand, integrity of sensor data originating from PCS networks must be guaranteed. At present, the only possibility considered secure consists in the application of a trusted third party for the generation and assignment of public key certificates for each PCS network participant. However, a trusted third party cannot guarantee user anonymity as there is at least one party which is able to map users, their messages, and their content to one and the same identity.

In order to avoid the need for a trusted third party, we consider a protocol extension for our P2P-based OSN architecture *Vegas* which allows for secure and privacy-preserving information querying inside PCS networks.

In the remainder of this paper section 2 gives a short overview of Vegas and details the proposed protocol extension. Section 3 discusses our extension focusing on performance, privacy, and security aspects. Work which is closely related to our approach is presented in section 4. Section 5 concludes the paper.

2 System Architecture

In the following, we review parts of our OSN architecture developed in previous work before we present our P2P-based overlay query routing extension which allows for privacy-preserving sensor task, subscription, and query distribution.

2.1 Vegas Design Principles

Our proposed P2P design Vegas builds on an architecture currently developed at the Ludwig-Maximilians-University Munich. The architecture focuses on privacy and security aspects within a P2P-based OSN. Vegas complies with a set of requirements that we consider inevitable for a secure and privacy preserving OSN. These requirements encompass a user's *informational self-determination*, *strong trust relationships* between friends, anywhere and anytime *profile availability*, and transparent *mobility support* [6].

Vegas represents a highly restrictive OSN as it does not support communication between participants that are not directly connected by an edge of the underlying social graph. This restriction is motivated by a problem we termed *social network pollution*. To give but a few examples of social network pollution, present OSNs offer the possibility for search operations on its social graphs, provide unsolicited friendship recommendations, and offer support for non-authorized linkage of a friend's friends. This leads to a multitude of unwanted friendship establishments i.e. links in the social graph which not necessarily represent a real friendship. Although this design choice disallows some appealing and beneficial applications, we consider this fact as an acceptable trade-off in order to guarantee a high degree of privacy and security.

2.2 Vegas Operation

In a nutshell, Vegas functions as follows: Any two users A and B who maintain a friendship in Vegas own a public key pair i.e. a *link-specific* key pair for each other. As user A holds a unique key pair $K^{-}_{A\to X_i}/K^{+}_{A\to X_i}(i \in 1,\ldots,n)$ for each of his n friends $X_1,...,X_n$, a key pair represents nothing else than a directed edge in the overall social graph. For the remainder of this paper, we always mean link-specific keys when we talk about keys and key pairs. The notion of a key $K^{[-|+]}_{A\to B}$ means that this key is a private/public key generated by A for exclusive communication with B. A utilizes B's public key $K^{+}_{B\to A}$ to encrypt messages intended for B. In addition, A's public key $K^{+}_{A\to B}$ is included into each message in order to map the originator of a message to its content. It should be stressed that, for the purpose

of signing, the application of A's private key $K^-_{A\to B}$ is restricted to messages addressed to B. Since a key pair represents an edge of the social graph, removal of such an edge – which equals to the cancellation of a friendship – is performed by the deletion of the matching key pair. This allows for easy key revocation as the application of keys is limited to two friends.

In order to allow for delay-tolerant offline communication, we build on an asynchronous message exchange scheme based on a concept presented in [13]. Vegas relies on well known services like email, SMS, or instant messaging which can be exploited to simulate the *exchanger* component. An exchanger represents the abstract concept of a message queue which may be used to transmit messages or any other kind of content. Any two Vegas friends A and B are aware of one or more such exchanger addresses of each other. In case A wants to send a message to B, A applies B's public key $K^+_{B\to A}$ to encrypt the message content. After signing the message with $K^-_{A\to B}$, A sends this message to one of B's exchangers. Now B can fetch this messages and identify sender A through his attached public key. It should be stressed that B is the only user that knows about the mapping of the included public key to the identity of user A. In case A considers the mapping of a public key to B's identity to be compromised, A can trigger a key refresh operation in order to replace all former key pairs shared with B.

2.3 Vegas Extension for Sensor Querying

Vegas represents a highly restrictive P2P OSN as it disallows communication between users that do not share an edge in the underlying social graph. Since P2P-based PCS applications require multi-hop (i.e multi-edge) communication we developed a protocol extension which allows for the application of Vegas for highly anonymous and trustful sensor information communication. In this paper we focus on the utilization of Vegas for sensor querying i.e. we provide an answer to the question *how to figure out one or more OSN users that a) support the discovery of sensor information, b) allow for sensor subscription, and c) facilitate sensor task placement.* Figure 1 illustrates this forwarding process in detail. Information from the OSN domain gives access to all properties and preferences from all OSN users that represent a friend. For instance, in case a user A wants to figure out other OSN users which could support querying sensor-related information, A can issue a sensor query to friends $X_1, \ldots, X_m$ ($1 \leq m \leq n$, n denotes the total number of A's friends) whose profiles match A's query content. Dependent on the kind of available sensors and user defined access policies a user X_i ($i \in 1, \ldots, m$) can respond to such a sensor query. To benefit from X_i's social relationships, we introduce a mechanism which allows for trusted and privacy-preserving forwarding of queries to users that do not share an edge with the query originator. Queries are tuples of the form `(operation, sensing properties, task properties)`. An example query for measuring the temperature in a certain area might look like this:

```
Q(i) = (put_task,(type=temp,frequency=1),(location={x,y},radius=r,time={t1,t2}))
```

At first, A attempts to locally match all profiles of his friends with the constraints of request `Q(i)`. Considering the request `Q(i)` a matchmaking process could be based on one or more heuristics that operate on the properties *location* and *time*. This restricts positive matches to users who not only support the requested sensor type and its properties but also visit places close to the provided location with a certain frequency and during certain periods of time. Since we do not limit requests to a predefined set of properties it could also be necessary to apply reasoning operations to reduce/increase the set of positive matches. After A finished the refinement process, he sends a request message `Q(i)` to all friends X_i ($i \in 1, \ldots, m$) whose profiles indicated a match. This message includes an exchanger addresses of A. A recipient,

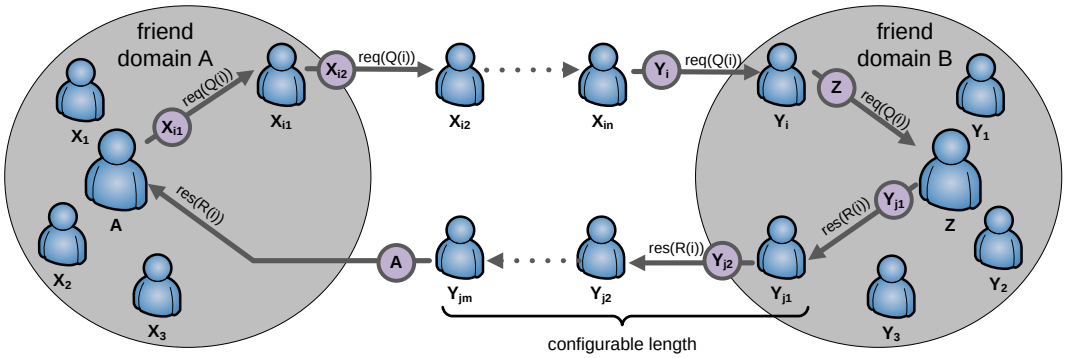

Figure 1 Schematic illustration of a query routing path. Messages are labeled with the corresponding exchanger utilized for transmission. Sending a message via exchanger T_i includes message encryption and signing with the corresponding key pairs.

in this example X_{i1}, who does not fulfill the requirements of `Q(i)` may decide to forward the query to all his friends whose profiles indicate a potential match for `Q(i)`. The decision whether to respond and forward a query depends on user preferences (either set by policies or manual interaction). This process repeats until `Q(i)` cannot be forwarded any longer or arrives at user Z who is able to setup the specified task. As we attach an exchanger address and a temporary public key of sender A to `Q(i)`, Z can send an encrypted response `R(i)` to A notifying him about a successful task setup. Due to our privacy demands, we enable Z to anonymize his identity by relying `R(i)` via one of his randomly selected friends Y_{j1}. To increase the degree of anonymity, we allow for a dynamic configuration of the number of edges `R(i)` has to traverse before it may reach the originator of the request. Dependent on the degree of anonymity, Y_{j1} does not directly respond to A but randomly selects one of his friends Y_{j2} as the next relay, and so far. Therefore, even multiple, successive responses are not delivered by the same user concealing a relation among those messages.

3 Discussion

Our approach represents a trade-off between support for efficient query routing inside PCS networks and the assurance of a high degree of anonymity and trust. This section provides some considerations regarding these aspects.

3.1 Routing Aspects

P2P networks can be separated into structured and unstructured systems. While unstructured systems suffer from query routing inefficiency due to their flooding approach, structured networks are susceptible to churn. The idea to infer query routing paths from OSN user profile information has recently been proposed for Pastry [10]. Our approach does not utilize the concept of DHTs but allows for simple forwarding based on preferences and capabilities of OSN participants. In essence, our design represents a trade-off since the application of OSN knowledge is expected to perform better than flooding but worse than DHT-based query routing. It should be mentioned that our proposed extension focuses on delay-tolerant networks. Although one could imagine scenarios which suffer from such a best effort approach, an asynchronous implementation is motivated by our demand for a high degree of anonymity and trust. In addition, this restriction can help to obtain a larger result set as users can forward requests even in case a recipient is currently offline.

3.2 Privacy Aspects

Although Vegas profiles are publicly available, by default, they are always encrypted with a friend's link-specific public key. Furthermore, all messages sent between two friends A and B are encrypted based on the corresponding public key. As long as direct messaging is limited to friends, Vegas adheres to all requirements listed in [6]. The proposed query extension for Vegas introduces a routing process which involves OSN users that do not share an edge with the query originator. However, our stringent privacy requirements are still fulfilled: Considering the query path of a sensor information request, the only information disclosed by the originator is one of probably multiple exchanger addresses. Dependent on the desired degree of anonymity, a sensing node may decide on the number of utilized exchangers as well as the frequency for exchanger refreshes. To increase the anonymity of a responding user, our extension allows him to specify the minimum number of edges a message has to pass before it can arrive at the originator of the request. Certainly, our extension cannot guard against protocol attacks like routing, storage, sybil, and eclipse attacks that all structured P2P networks suffer from. For instance, it takes no effort to replace an exchanger address to forward responses to another user. However, assuming trustful friendships, our extension cannot be exploited for security attacks targeting at message integrity or disclosure of user identities.

3.3 Trust Aspects

Vegas does not support unsolicited friendship establishment. Due to the introduction of multi-edge query routing, this stringent restriction becomes relaxed as private information disseminates over the boundary of the personal friendship domain. However, even in case a user observes hostile behavior of his friends, it necessitates a simple key pair deletion to cancel any association with such a friend. The degree of trust achieved by our design is not equivalent to that offered by a single trusted third party. However, compared to the application of a *web of trust* solution, our extension allows for much better trust relations between users that do not share an edge in the social graph. In summary, our trust model must be considered as a serious alternative to a trusted third party solution as it achieves complete informational self-determination in a decentralized way.

4 Related Work

A lot of research has been published with concepts focusing on privacy, security, and integrity aspects in PCS networks. The key idea of PriSense [11] is to slice data and send it via several different nodes, instead of sending it at once and directly to a server. Thereby, it conceals the data sensed by the individual users; however, it is still known which users participated. Our solution conceals the identities of participating users to allow for a real anonymous participation. Another approach is to add noise to sensed data, which can be subtracted in a later stage. In PoolView [7] measured time-series data is superimposed with a noise model, which is shared within the community. It preserves recently sensed values as well as their changes with time. Instead of perturbing measurements, AnonySense [3] employs several anonymization components, such as Tor and MIX networks, to obscure a user's identity. k-anonymity is ensured through a registration authority. An orthogonal approach is to actively engage users to decide on *what to reveal to whom*, and to learn what kind of data can be shared without compromising the user's privacy [12]. However, this concept simply retains sensitive data which results in a smaller amount of useful information.

Due to our fully anonymous communication, our solution also allows for sharing of sensitive data. An important research field represents privacy protection in OSNs. Some related work already proposes the application of P2P concepts to allow for privacy-preserving social networking [4, 2]. OSNs have also been combined with PCS networks but merely to enable new applications [8, 9] and not to enhance privacy. To the best of our knowledge, our approach is the first that uses an OSN to ensure privacy and trust in PCS networks.

5 Conclusion and Future Work

We proposed a protocol extension to our OSN design Vegas which achieves a high degree of anonymity and trust for delay-tolerant PCS networks. Our solution does not rely on a trusted third party but achieves trust by exploiting knowledge about the social graph. As our extension circumvents the spread of social network pollution and therefore complies with our requirement for perfect informational self-determination, we achieve a higher degree of trust than with a decentralized solution like the web of trust. In our future work we intend to perform several user studies to gain knowledge about the applicability of our approach in real world settings. In parallel, we will conduct intense simulations of typical protocol attacks identify the requirements for important system parameters like the minimum number of friends necessary to achieve a sufficient degree of anonymity and trust.

References

1 A. Beach, M. Gartrell, X. Xing, R. Han, Q. Lv, S. Mishra, and K. Seada. Fusing mobile, sensor, and social data to fully enable context-aware computing. In *Proc. of HotMobile '10*, pages 60–65. ACM, 2010.

2 S. Buchegger, D. Schiöberg, L. H. Vu, and A. Datta. PeerSoN: P2P social networking - early experiences and insights. In *Proc. of SocialNet '09*, 2009.

3 C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. AnonySense: Privacy-aware people-centric sensing. In *Proc. of MobiSys '08*, pages 211–224. ACM, 2008.

4 L. A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *Proc. of WONS'09*, pages 133–140. IEEE, 2009.

5 T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. PRISM: platform for remote sensing using smartphones. In *Proc. of MobiSys '10*, pages 63–76. ACM, 2010.

6 M. Dürr, M. Werner, and M. Maier. Re-Socializing Online Social Networks. In *Proc. of CPSCom'10*. IEEE, Dec 2010.

7 R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher. Poolview: stream privacy for grassroots participatory sensing. In *Proc. of SenSys '08*, pages 281–294. ACM, 2008.

8 I. Krontiris and F.C. Freiling. Integrating people-centric sensing with social networks: A privacy research agenda. In Proc. of SESOC 2010, 2010.

9 E. Miluzzo, N. Lane, S. Eisenman, and A. Campbell. CenceMe – Injecting Sensing Presence into Social Networking Applications. In *Smart Sensing and Context*, volume 4793 of *LNCS*, pages 1–28. Springer Berlin / Heidelberg, 2007.

10 A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot. MobiClique: middleware for mobile social networking. In *Proc. of WOSN '09*, pages 49–54. ACM, 2009.

11 J. Shi, R. Zhang, Y. Liu, and Y. Zhang. Prisense: privacy-preserving data aggregation in people-centric urban sensing systems. In *Proc. INFOCOM'10*, pages 758–766. IEEE, 2010.

12 K. Shilton. Four billion little brothers?: privacy, mobile phones, and ubiquitous data collection. *Commun. ACM*, 52(11):48–53, 2009.

13 M. Werner. A privacy-enabled architecture for location-based services. In *Proc. of MobiSec '10*, 2010.